CLAIMS:

1.   A method for identifying non-externalized strings that are not hard-coded comprising the steps of:

      scanning a code for a first pair of string delimiters; and

      determining whether a string within said first pair of string delimiters is a path name to a resource file;

      wherein if said string is a path name to said resource file then said string is a non-externalized string that is not hard-coded.

2.   The method as recited in claim 1 further comprising the step of:

      identifying said string as a possible hard-coded string if said string is not a path name to said resource file.

3.   The method as recited in claim 1, wherein said code comprises Java code.

4.   The method as recited in claim 1, wherein said path name is a uniform resource locator.

5.   The method as recited in claim 1, wherein said resource file is a resource bundle.

1
2
3

6.   The method as recited in claim 1, wherein said string within said first pair of string delimiters is a path name to said resource file if said string is in a dot delimited notation.

1
2

7.   The method as recited in claim 1, wherein said code is scanned line by line until said first pair of string delimiters is identified.

8.   The method as recited in claim 7, wherein if there is any more code to be scanned after said first pair of string delimiters is identified, then the method further comprises the step of:

          continuing to scan said code for a second pair of string delimiters.

9.   A computer program product in a computer readable medium for identifying non-externalized strings that are not hard-coded, comprising:

        programming operable for scanning a code for a first pair of string delimiters; and

        programming operable for determining whether a string within said first pair of string delimiters is a path name to a resource file;

        wherein if said string is a path name to said resource file then said string is a non-externalized string that is not hard-coded.

10.   The computer program product as recited in claim 9 further comprises:

        programming operable for identifying said string as a possible hard-coded string if said string is not a path name to said resource file.

11.   The computer program product as recited in claim 9, wherein said code comprises Java code.

12.   The computer program product as recited in claim 9, wherein said path name is a uniform resource locator.

13.   The computer program product as recited in claim 9, wherein said resource file is a resource bundle.

14.   The computer program product as recited in claim 9, wherein said string within said first pair of string delimiters is a path name to said resource file if said string is in a dot delimited notation.

1   15.   The computer program product as recited in claim 9, wherein said code is

2   scanned line by line until said first pair of string delimiters is identified.

1   16.   The computer program product as recited in claim 15, wherein if there is any

2   more code to be scanned after said first of string delimiters is identified, then the

3   computer program product further comprises:

        programming operable for continuing to scan said code for a second pair of string

delimiters.

1   17.   A data processing system, comprising:

2   a processor; and

3   a memory unit for storing instructions of said processor;

4   an input mechanism;

5   an output mechanism;

6   a bus system for coupling the processor to the memory unit, input mechanism,

7   and output mechanism;

8   means for scanning a code for a first pair string delimiters; and

9   means for determining whether a string within said first pair of string delimiters

10   is a path name to a resource file;

11   wherein if said string is a path name to said resource file then said string is a

12   non-externalized string that is not hard-coded.

        18.   The data processing system as recited in claim 17, wherein the system further

        comprises:

                means for identifying said string as a possible hard-coded sting if said string is

4   not a path name to said resource file.

1   19.   The data processing system as recited in claim 17, wherein said code comprises

2   Java code.

1   20.   The data processing system as recited in claim 17, wherein said path name is a

2   uniform resource locator.

1   21.   The data processing system as recited in claim 17, wherein said resource file is

2   a resource bundle.

1   22.   The data processing system as recited in claim 17, wherein said string within said

2   first pair of string delimiters is a path name to said resource file if said string is in a dot

3   delimited notation.

23.   The data processing system as recited in claim 17, wherein said code is scanned

line by line until said first pair of string delimiters is identified.

24.   The data processing system as recited in claim 23, wherein if there is any more

code to be scanned after said first pair of string delimiters is identified, then the system

further comprises:

        means for continuing to scan said code for a second pair of string delimiters.